DATA ANALYSIS ON E- STORE

Project Author: Neha Kunwar Solanki

OBJECTIVE:

In this project, we are working on E-commerce Data to get some information so that the information can be used for analytical purpose and decision making, useful for Maximizing Business Profit. Huge data sets will help Organizations to address potential customers in a meaningful way.

Dataset information that could be used for future decisions, improve customer engagement so that newly launch product information can be shared to them.

SAMPLE DATA SETS FOR ANALYSIS:

1. **Customer:** File Customer.txt

Customer_id	First name	last name	age	Address

2. Transaction: File used txns-large.txt

Date	uid	amount	category	product	city	state	Payment

PROJECT DESCRIPTION -

We are provided with certain use-cases to get the required data. For all the use-cases we will be using a Map-reduce approach. The Map Reduce Approach totally works on Key-Value pair as Input and Output. There will be a Driver Class, Mapper Class and a Reducer Class.

TECHNOLOGY USED:

- Apache Hadoop
- Map-Reduce Programming in java.
- PIG
- HIVE

SOFTWARE USED:

- Eclipse IDE
- Oracle Virtual Machine
- Ubuntu
- JDK 1.7

USE CASES

1. Categorization of customer based on Amount Scenario:

The system keeps track of different customer's information by their unique code. Whenever user purchases a product of a particular price or within range of amount than at time the user will provide with similar type of product within the same range.

- **Find all the transaction where amt>160**
  Validation Constraints: Yes, If the user is passing String in place of number we will be displayed an error message to provide valid input and start the job again.

  Output: Using Custom input

```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$
```

```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom12
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
190
16/11/21 13:49:40 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 13:49:41 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool int
```

HIVE Output for same task

```
hive> select tid from transaction where amt>160;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201611222318_0176, Tracking URL = http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201611222318_0176
```

```
00049955
00049956
00049960
00049968
00049973
00049978
00049980
00049981
00049991
00049994
00049996
00049998
00049999
Time taken: 71.457 seconds
hive>
```

PIG Output for same task

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate uid, amt;
step3 = FILTER step2 by amt>160;
DUMP step3;
```

```
(4004939,198.32)
(4002061,175.61)
(4004311,184.18)
(4008449,192.67)
(4004318,199.07)
(4008637,198.4)
(4003685,191.29)
(4005772,177.22)
(4007287,163.81)
(4007843,180.41)
(4001406,168.49)
[cloudera@localhost Desktop]$
```

2. Customer Price Based transaction information

In this use case, we are finding all the transaction where amount is more than 170 and less than 200 paid by customer. So that we come to know about the specific amount or more than that is paid by customer for purchasing and new product and services in same rang can be shared to them.

- **Count all the transaction where amount is between 175 to 200**

1) **Validation Constraints: Yes,** We will be accepting user input for minimum and maximum limit for price. The maximum price can't be less than minimum price. If attempted a message will be displayed for this**.** And also tells the user to run the task again with proper inputs.

   Minimum amount cannot be less than 0.Error message

   Maximum amount cannot be less than 0. Error message

   Output: <u>Using Custom input</u>

```
hduser@ubuntu64server: ~
                Reduce input groups=1
                Reduce shuffle bytes=36835
                Reduce input records=2833
                Reduce output records=1
                Spilled Records=5666
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=239
                CPU time spent (ms)=2260
                Physical memory (bytes) snapshot=300953600
                Virtual memory (bytes) snapshot=3754459136
                Total committed heap usage (bytes)=137498624
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4418139
        File Output Format Counters
                Bytes Written=55
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/hee/part-r-00000
Total number of transaction for your search are : 2833
```

Hive Output for the same task:

```
hive> select count(*) from transaction where amt>145 and amt<156;

Total MapReduce CPU Time Spent: 3 seconds 910 msec
OK
2833
Time taken: 42.222 seconds
hive>
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate tid, amt, uid;
step3 = FILTER step2 by amt >175;
step4 = FILTER step3 by amt <200;
step5 = FOREACH step4 generate 1 as one;
step6 = GROUP step5 by one;
step7 = FOREACH step6 generate COUNT(step5.one);
DUMP step7;
```

```
2016-11-24 01:39:47,048 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2016-11-24 01:39:47,049 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process
(2833)
[cloudera@localhost Desktop]$
```

3. Overall Transaction counting for each user:

In this use case, we will fetch a Customer Overall Transactional report against each customer ID, Counting will be done for the no. of transactions. Also for each customer, all the transaction amount is added. Finally we will display the count and total transaction amount for every customer.

- **Calculate the total sum and total count of all the transaction for each user id**
  Output: <u>Using Custom input&</u> <u>Validation Constraints</u>

```
hduser@ubuntu64server: ~
4009957 142.57   4
4009958 471.94   4
4009959 142.1   1
4009960 642.1100000000001   6
4009961 877.32   7
4009962 419.83   5
4009963 161.82999999999998   2
4009964 386.46999999999997   5
4009965 145.1   1
4009966 412.84   4
4009967 622.6800000000001   6
4009968 704.07   8
4009969 461.19   5
4009970 154.92   1
4009971 528.3399999999999   5
4009972 691.19   9
```

Hive Output for the same task:

```
hive> select uid, sum(amt) , count(tid) from transaction group by uid;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1

4009990 754.4200000000001       7
4009991 372.45  3
4009992 336.73  3
4009993 331.90000000000003      3
4009994 461.03999999999996      4
4009995 455.13  7
4009996 836.1200000000001       8
4009997 486.18999999999994      4
4009998 665.7   6
4009999 682.0200000000001       8
Time taken: 59.775 seconds
hive>
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate tid, uid, amt;
step3 = GROUP step2 by uid;
step4 = FOREACH step3 GENERATE group, COUNT (step2.tid), SUM(step2.amt);
DUMP step4;
```

```
(4009987,5,516.98)
(4009988,2,234.05)
(4009989,2,200.95)
(4009990,7,754.4200000000001)
(4009991,3,372.45)
(4009992,3,336.73)
(4009993,3,331.90000000000003)
(4009994,4,461.03999999999996)
(4009995,7,455.13)
(4009996,8,836.1200000000001)
(4009997,4,486.19000000000005)
(4009998,6,665.7)
(4009999,8,682.0200000000001)
[cloudera@localhost Desktop]$
```

4. Calculate the average transaction value for each user id

In this case we will fetch the transaction amount data of each user and get the average of all transaction as per individual user id. Average rating for each user can be done periodically for analysis. Average transaction only decides weekly, monthly and yearly product services.

- **Calculate the total sum and total count of all the transaction for each user id**

  Validation Constraints: Yes

  Output: **<u>Using Custom input</u>**

```
hduser@ubuntu64server:~$ hadoop fs -ls /gan6
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 21:05 /gan6/_SUCCESS
-rw-r--r--   1 hduser supergroup         74 2016-11-21 21:05 /gan6/part-r-00000
hduser@ubuntu64server:~$ hadoop fs -cat /gan6/part-r-00000
4009775 Sum : 541.6400000000001   Count : 7   Average : 77.37714285714287
```

Pig Output for the same:

```
a = load '/user/cloudera/Transactional.dat' using PigStorage(',') as (tid, d ,
uid, amt:double, cat, prod, city, state, pt);
b = foreach a generate uid, amt;
c = group b by uid;
d = foreach c generate group, COUNT(b.amt);
dump d;
```

```
(4009977,400.78)
(4009978,106.42)
(4009979,785.28)
(4009980,567.1199999999999)
(4009981,395.14)
(4009982,325.23)
(4009983,342.75000000000006)
(4009984,522.66)
(4009985,430.03000000000003)
(4009986,230.87)
(4009987,516.98)
(4009988,234.05)
(4009989,200.95)
(4009990,754.4200000000001)
(4009991,372.45)
(4009992,336.73)
(4009993,331.90000000000003)
(4009994,461.03999999999996)
(4009995,455.13)
(4009996,836.1200000000001)
(4009997,486.19000000000005)
(4009998,665.7)
(4009999,682.0200000000001)
[cloudera@localhost ~]$
```

5. Division of single file into multiple files

In this use case dataset is divided into multiple sub file according to product category. As a developers, fetching data from two table and divide them according to product category. For example how many customer has used credit card for as a payment mode, How many customer took offer on products so that when season comes for discount we can inform those customer about discount offer.

- **Divide the file into 12 files, each file containing each month of data. For eg. file 1 should contain data of January txn, file 2 should contain data of feb txn.**

Validation Constraints: Yes

Output: <u>Using Custom input</u>

```
hduser@ubuntu64server:~$ hadoop fs -la   /uio
-la: Unknown command
hduser@ubuntu64server:~$ hadoop fs -ls   /uio
Found 13 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 22:30 /uio/_SUCCESS
-rw-r--r--   1 hduser supergroup     377449 2016-11-21 22:28 /uio/part-r-00000
-rw-r--r--   1 hduser supergroup     339311 2016-11-21 22:28 /uio/part-r-00001
-rw-r--r--   1 hduser supergroup     385895 2016-11-21 22:28 /uio/part-r-00002
-rw-r--r--   1 hduser supergroup     368421 2016-11-21 22:28 /uio/part-r-00003
-rw-r--r--   1 hduser supergroup     371798 2016-11-21 22:28 /uio/part-r-00004
-rw-r--r--   1 hduser supergroup     368247 2016-11-21 22:28 /uio/part-r-00005
-rw-r--r--   1 hduser supergroup     375554 2016-11-21 22:29 /uio/part-r-00006
-rw-r--r--   1 hduser supergroup     374305 2016-11-21 22:29 /uio/part-r-00007
-rw-r--r--   1 hduser supergroup     367955 2016-11-21 22:29 /uio/part-r-00008
-rw-r--r--   1 hduser supergroup     368733 2016-11-21 22:29 /uio/part-r-00009
-rw-r--r--   1 hduser supergroup     353858 2016-11-21 22:29 /uio/part-r-00010
-rw-r--r--   1 hduser supergroup     366614 2016-11-21 22:29 /uio/part-r-00011
hduser@ubuntu64server:~$ [2~^[[2~
```

Hive Output for the same task:

```
hive> select * from transaction where substr(d,0,2)=01;
Total MapReduce jobs = 1
Launching Job 1 out of 1
```

```
00049914        01-14-2015      4007397 83.47   Outdoor Play Equipment  Outdoor Playsets        Montgomery      Alabama credit
00049933        01-02-2015      4006816 17.37   Combat Sports   Martial Arts    Des Moines      Iowa    credit
00049959        01-21-2015      4006137 28.39   Exercise & Fitness      Free Weights    Sacramento      California      cash
00049962        01-01-2015      4002152 58.55   Water Sports    Kitesurfing     San Diego       California      credit
00049973        01-27-2015      4004311 184.18  Outdoor Recreation      Running Coral Springs   Florida credit
00049974        01-22-2015      4001002 20.71   Team Sports     Rugby   Vancouver       Washington      cash
00049994        01-05-2015      4005772 177.22  Outdoor Recreation      Archery Baltimore       Maryland        credit
Time taken: 93.123 seconds
hive>
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate SUBSTRING(d,0,2) as month;
step3 = GROUP step2 by month;
step4 = filter step3 by group=='01';
STORE step1 INTO '/user/cloudera/part-00001';
step4 = filter step3 by group=='02';
STORE step1 INTO '/user/cloudera/part-00002';
step4 = filter step3 by group=='03';
STORE step1 INTO '/user/cloudera/part-00003';
step4 = filter step3 by group=='04';
STORE step1 INTO '/user/cloudera/part-00004';
step4 = filter step3 by group=='05';
STORE step1 INTO '/user/cloudera/part-00005';
step4 = filter step3 by group=='06';
STORE step1 INTO '/user/cloudera/part-00006';
step4 = filter step3 by group=='07';
STORE step1 INTO '/user/cloudera/part-00007';
step4 = filter step3 by group=='08';
STORE step1 INTO '/user/cloudera/part-00008';
step4 = filter step3 by group=='09';
STORE step1 INTO '/user/cloudera/part-00009';
step4 = filter step3 by group=='10';
STORE step1 INTO '/user/cloudera/part-00010';
step4 = filter step3 by group=='11';
STORE step1 INTO '/user/cloudera/part-00011';
step4 = filter step3 by group=='12';
STORE step1 INTO '/user/cloudera/part-00012';
```

6. The profession of user who has spent the maximum amount

In this use case we are given a task to find the name of profession from customer dataset to find maximum amount. Next new products marketing starts from him by giving discount of 20% on purchasing

- **Find the profession of user who has spent the maximum amount**

  Validation Constraints: Yes

  Output: **<u>Using Custom input</u>**

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive30/part-r-00000
Pilot   1700.17
```

Hive Output for the same task:

```
hive> select fname ,sum(amt) tamt from customer a join transaction b on a.uid=b.uid group by fname order by tamt desc limit 3;
```

```
OK
Pilot   1700.1700000000005
Time taken: 189.081 seconds
hive>
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid,fname,lname,age:double,prof);
step3 = JOIN step1 by uid, step2 by custid;
step4 = GROUP step3 by prof;
step5 = FOREACH step4 GENERATE group, SUM(step3.amt)as tamt;
step6 = ORDER step5 by tamt desc;
step7 = LIMIT step6 1;
dump step7;
```

```
2016-11-24 03:32:02,484 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths
2016-11-24 03:32:02,484 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total inpu
(Pilot,1700.17)
[cloudera@localhost Desktop]$
```

7. **New Product and Services:**

In this use case, we are finding three top spenders report to whom organization can offer new launching Services like yoga products, gym products, Air sports, life jackets etc.

- **Find the name of top 3 spenders.**

  Output: <u>Using Custom input</u>

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive31/part-r-00000
Karen   1080.42
Kristina        980.51
Elsie   719.66
```

Hive Output for the same task:

```
hive> select fname ,sum(amt) tamt from customer a join transaction b on a.uid=b.uid group by fname order by tamt desc limit 3;

Karen   1080.4199999999998
Kristina        980.51
Elsie   719.66
Time taken: 158.986 seconds
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid,fname,lname,age:double,prof);
step3 = JOIN step1 by uid, step2 by custid;
step4 = GROUP step3 by fname;
step5 = FOREACH step4 GENERATE group, SUM(step3.amt)as tamt;
step6 = ORDER step5 by tamt desc;
step7 = LIMIT step6 3;
dump step7;

(Karen,1080.42)
(Kristina,980.51)
(Elsie,719.66)
[cloudera@localhost Desktop]$
```

8. Retaining Customers: Customer lifetime value

In this use case, searching particular customer who has made highest transaction so that we can analyze number of unique purchase mode, average price of products, average price or orders And number of days and session leading to a transaction.

- **Find the profession of user who has spend the maximum amount**

  Output: **<u>Using Custom input</u>**

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive30/part-r-00000
Pilot   1700.17
```

Hive Output for the same task:

```
hive> select prof ,sum(amt) tamt from customer a join transaction b on a.uid=b.uid group by prof order by tamt desc limit 1;
```

```
OK
Pilot   1700.1700000000005
Time taken: 189.081 seconds
hive>
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid,fname,lname,age:double,prof);
step3 = JOIN step1 by uid, step2 by custid;
step4 = GROUP step3 by prof;
step5 = FOREACH step4 GENERATE group, SUM(step3.amt)as tamt;
step6 = ORDER step5 by tamt desc;
step7 = LIMIT step6 1;
dump step7;
```

```
2016-11-24 03:32:02,484 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths
2016-11-24 03:32:02,484 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total inpu
(Pilot,1700.17)
[cloudera@localhost Desktop]$
```

9. Special rewards: Extra point Events

In this use case, fetching all customer information in July month to give them offer like Extra point events. An extra point's event is a great way to boost program engagement and encourage shoppers to spend points

- **Find the user who has spent the max amount in July month**

  Output: <u>Using Custom input</u>

```
hduser@ubuntu64server:~$ hadoop fs -cat /Olive32/part-r-00000
Karen   155.18
```

Hive Output for the same task:

```
hive> select fname ,sum(amt) tamt  from customer a join transaction b on a.uid=b.uid where substr(d,0,2)=07 group by fname order by tamt
desc limit 1;

Karen   155.18
Time taken: 154.814 seconds
```

Pig Output For the same task:

```
step1 = LOAD '/user/cloudera/Transactional.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid, fname, lname, age:double, prof);
step3 = JOIN step1 by uid , step2 by custid;
step4 = FOREACH step3 GENERATE fname, SUBSTRING(d,0,2) as mon, amt;
step5 = FILTER step4 by mon=='07';
step6 = GROUP step5 by fname;
step7 = FOREACH step6 GENERATE group, SUM(step5.amt) as tcnt;
step8 = ORDER step7 by tcnt desc;
step9 = LIMIT step8 1;
dump step9;
```

```
(Karen,155.18)
[cloudera@localhost Desktop]$
```

CONCLUSION: Analysis done on relevant very large sets of data for Statistical analysis, data mining, predictive analytics, and text mining. And we built a summary table to aggregate the detail at a monthly level transaction, a table to aggregate the detail at a year-to-date level, and a final summary table for the division level.